Article

PEOPLE COUNTING USING CONVOLUTIONAL NEURAL NETWORK

Do Ngoc Hieu*

** Faculty of Mechanical Engineering, Ho Chi Minh City university of Technology and Education, Thu Duc city, Viet Nam, dnh02.cv@gmail.com*

Published: 22 May 2023

Keywords: Convolutional Neural Network, Human Detection, OpenCV, Streamlit

Abstract:
This study introduces one of the most popular methods for counting people in the images, using Convolutional Neural Network. By using Deep Learning techniques to overcome challenges such as lighting changes and obstructions. The model has multiple layers for extracting features and classifying, can detect and count individuals accurately in images. Experimental results on standard datasets show that the method performs better than traditional counting methods. This CNN-based people counting method has potential for real-world applications in monitoring crowds.

I. INTRODUCTION

In this project, I employed Convolutional Neural Network (CNN) technique to train a model capable of dectecting whether an object is Human or not. The training process took place within the Google Colab environment to enhance the model's performance.

Additionally, I used OpenCV to crop these objects using the Contour function. Following the detection stage, the model simply counts the number of "Human" objects identified. To provide a user-friendly experience, I implemented a simple interface using the Streamlit library, ensuring ease of use for all users.

II. METHODOLOGY

1. Data Collection:
The first step in this project is gathering a diverse dataset comprising images containing both human and non-human objects.

2. CNN Model Training:
The Convolutional Neural Network (CNN) method was selected as the primary approach for training the object detection model. The collected dataset was split into training and testing sets to evaluate the model's performance accurately.

3. Google Colab Environment:
To optimize the model's performance, the training process was conducted in the Google Colab environment. This platform provides powerful GPUs and resources, allowing for faster training iterations

4. Preprocessing with OpenCV:
The OpenCV library was employed for preprocessing tasks, specifically for object cropping. The Contour function in OpenCV was utilized to extract individual objects from the input images. This step was employed on Visual Studio Code to obtain the high performace.

5. Object Detection and Counting:
After preprocessing, the trained CNN model was applied to the cropped objects to determine whether each object was human or non-human. The model's predictions were used to count the number of "Human" objects present in the images. This step provided valuable insights into the composition of the dataset and allowed for further analysis.

6. User Interface Development:
To create an intuitive and user-friendly experience, a simple interface was developed using the Streamlit library. Streamlit offers an easy-to-use framework for building interactive web applications with minimal effort. The interface allowed users to input images and receive real-time predictions on whether the objects in the images were human or not.

Overall, this methodology encompassed data collection, CNN model training, utilization of the Google Colab environment for optimization, preprocessing with OpenCV, object detection and counting, and the development of a user-friendly interface using the Streamlit library. These steps ensured an effective and efficient approach to detecting human objects in images.

III. MODEL AND ALGORITHM

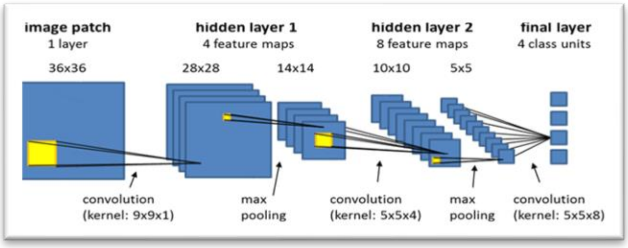


Fig 1. CNN Algorithm Diagram

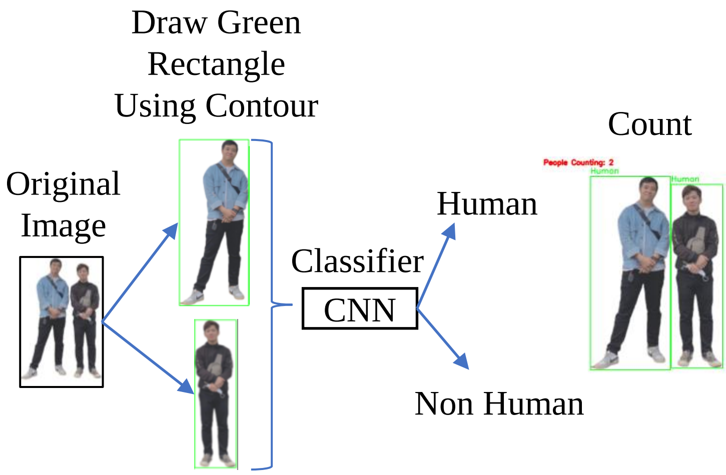
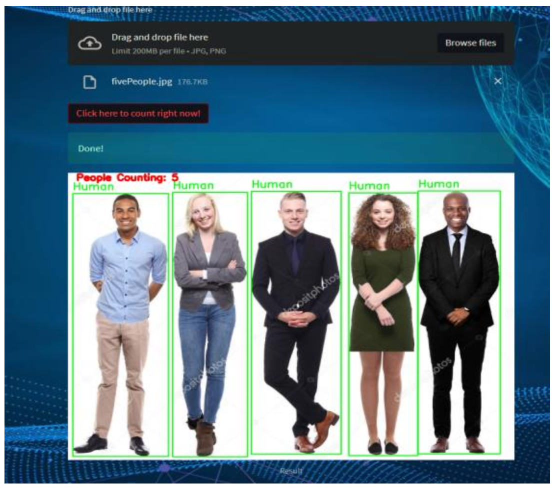


Fig 2. Model Algorithm Diagram

IV. RESULT AND DISSCUSION

Result displays on Streamlit web:



Disscusion:

One of the main advantages of employing the CNN method is its ability to automatically learn features from the input data, making it well-suited for image classification tasks. By training the model on a diverse dataset containing human and non-human objects. But the method relies on counting the number of humans through edge detection using the Contour function. This imposes complexities on the input images, requiring medium-sized images with a white background and individuals spaced apart. These limitations restrict the model's versatility and may lead to inaccurate results without careful consideration of image quality and composition.

To improve the model, we can focus on expanding the dataset, implementing data increasing techniques, and exploring advanced algorithms with higher accuracy. Base on Deep Learning techniques, refining preprocessing methods, and fine-tuning hyperparameters can also enhance the model's performance. These developments aim to overcome limitations and achieve better efficiency and accuracy in human object detection, while reducing requires related to input image requirements.

V. CONCLUSION

In conclusion, the project aimed to develop a model for detecting human objects using the CNN method. The methodology involved data collection, CNN model training, utilization of the Google Colab environment, OpenCV preprocessing, object detection, and the development of a user interface using the Streamlit library.

However, there were drawbacks in relying on contour-based counting and the limitations imposed by image requirements. To overcome these limitations, potential developments were suggested, including enhancing model training, exploring advanced algorithms, leveraging deep learning techniques, refining preprocessing methods, and fine-tuning hyperparameters. These developments aim to improve the model's accuracy and efficiency in human object detection while reducing constraints related to input image requirements. Overall, this project lays the foundation for further research and advancements in human object detection algorithms.

REFERENCES

1. Introduction To CNN:
https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/

2. Human Detection And Tracking:
https://link.springer.com/referenceworkentry/10.1007/978-0-387-73003-5_35#:~:text=Human%20detection%20is%20the%20task,templates%20or%20patterns%20of%20people.

3. Detect An Object With OpenCV-Python:
https://www.geeksforgeeks.org/detect-an-object-with-opencv-python/

4. Create An App With Streamlit:
https://docs.streamlit.io/library/get-started/create-an-app